IN THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) An apparatus for computer hardware multithreading comprising:

a plurality of processors, each processor having hardware support for the capability of executing a plurality of threads;

a memory coupled to the plurality of processors; and

a thread dispatch mechanism residing in the memory and executed by at least one of the plurality of processors, the thread dispatch mechanism determining which of the plurality of processors are idle ~~when the processor is able to accept a new thread for each processor hardware threads~~, which of the plurality of processors is busy processing a thread but can accept an additional thread, and which of the plurality of processors cannot accept an additional thread since it is working on a maximum number of threads the processor can execute and, the thread dispatch mechanism dispatching a new thread to an idle processor, if one exists.

2. (Original) The apparatus of claim 1 wherein, if none of the plurality of processors is idle and if at least one of the plurality of processors can accept an additional thread, the thread dispatch mechanism dispatches the new thread to one of the plurality of processors that can accept an additional thread.

3. (Original) The apparatus of claim 1 wherein, if all of the plurality of processors cannot accept an additional thread, the thread dispatch mechanism waits for one of the plurality of processors to complete processing a thread, thereby becoming a processor that can accept an additional thread, and then dispatches the thread to the processor that can accept an additional thread.

4. (Currently Amended) A method for dispatching threads in a computer system that includes a plurality of processors that can each support hardware multithreading to execute a plurality of threads, the method comprising the steps of:

(1) determining the status of each of the plurality of processors, wherein a processor is idle if not executing any threads ~~when able to accept a new thread for each processor hardware thread~~, wherein the processor can accept an additional thread if busy working on one or more threads but has the capacity to process the additional thread, and wherein the processor cannot accept an additional thread if busy working on a maximum number of threads the processor can execute; and

(2) dispatching a new thread to an idle processor, if one exists.

5. (Original) The method of claim 4 further comprising the step of:

if none of the plurality of processors is idle and if at least one of the plurality of processors can accept an additional thread, the thread dispatch mechanism dispatches the new thread to one of the plurality of processors that can accept an additional thread.

6. (Original) The method of claim 4 further comprising the steps of:

if all of the plurality of processors cannot accept an additional thread, the thread dispatch mechanism waits for one of the plurality of processors to complete processing a thread, thereby becoming a processor that can accept an additional thread, and then dispatches the thread to the processor that can accept an additional thread.

7. (Currently Amended) A program product comprising:

(A) a thread dispatch mechanism that determines which of a plurality of processors in a hardware multithreading, multiprocessor computer system are idle ~~when the processor is able to accept a new thread for each processor hardware thread~~, which of the plurality of processors is busy but can accept an additional thread, and which of the plurality of processors cannot accept an additional thread since it is working on a maximum number of threads the processor can execute, the thread dispatch mechanism dispatching a new thread to an idle processor, if one exists, wherein each processor can execute a plurality of threads; and

(B) computer-readable signal bearing media bearing the thread dispatch mechanism.

8. (Original) The program product of claim 7 wherein the computer-readable signal bearing media comprises recordable media.

9. (Original) The program product of claim 7 wherein the computer-readable signal bearing media comprises transmission media.

10. (Original) The program product of claim 7 wherein, if none of the plurality of processors is idle and if at least one of the plurality of processors can accept an additional thread, the thread dispatch mechanism dispatches the new thread to one of the plurality of processors that can accept an additional thread.

11. (Original) The program product of claim 7 wherein, if all of the plurality of processors cannot accept an additional thread, the thread dispatch mechanism waits for one of the plurality of processors to complete processing a thread, thereby becoming a processor that can accept an additional thread, and then dispatches the thread to the processor that can accept an additional thread.

12. (Previously added) The apparatus of claim 1 wherein all processors are made busy with a first thread before dispatching a second thread to any processor.

13. (Previously added) The method of claim 4 wherein all processors are made busy with a first thread before dispatching a second thread to any processor.

14. (Previously added) The program product of claim 7 wherein all processors are made busy with a first thread before dispatching a second thread to any processor.